



新浪微博：C语言图书
腾讯QQ：4006751066

C语言学习路线图



C语言函数 参考手册

快速服务： 微博、QQ在线服务

自学视频： 72集大型多媒体自学视频

海量资源： 模块库、案例库、素材库、题库



明日科技 编著

本书提供了内容丰富的配套资源，可以登录www.tup.com.cn，找到本书后，在该页面的“网络资源”超链接处下载。也可以访问本书的新浪微博，根据提示链接下载。

清华大学出版社

C 语言学习路线图

C 语言函数参考手册

明日科技 编著

（模块库、案例库、素材库、题库）

（微博、QQ、论坛技术支持）

清华大学出版社

北 京

内 容 简 介

本书是一本 C 语言函数查询参考手册, 包含了 C 语言程序开发中应用的各种函数。全书共 20 章, 按照 C 语言库函数头文件进行分章, 每个头文件为一章, 在每一个头文件中以字母的顺序对该头文件中的各个函数进行排序, 其中每个函数都按照功能、语法来介绍, 在每个方法后都有相关的示例。

本书提供了源程序、素材, 提供了相关的模块库、案例库、素材库、题库等多种形式的辅助学习资料, 还提供迅速及时的微博、QQ、论坛等技术支持。

本书内容详尽、示例丰富, 非常适合作为编程人员及项目开发人员的工具用书。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

C 语言函数参考手册/明日科技编著. —北京: 清华大学出版社, 2012.2
(C 语言学习路线图)

ISBN 978-7-302-27664-7

I. ①C… II. ①明… III. ①C 语言-程序设计-参考资料 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2011) 第 267866 号

责任编辑: 赵洛育 刘利民

版式设计: 文森时代

责任校对: 姜 彦

责任印制:

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:

装 订 者:

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 24.75 字 数: 572 千字

版 次: 2012 年 2 月第 1 版 印 次: 2012 年 2 月第 1 次印刷

印 数: 1~5000

定 价: 49.80 元

产品编号: 045370-01

前言

Preface



C 语言作为当前程序开发中最流行的编程语言之一，以其简单、高效、快捷、方便的特性吸引了大批用户。根据笔者使用 C 语言的经验，相信对 C 语言程序开发人员来说，C 语言函数参考手册是必需的。因此，笔者编写了本书，希望能真正帮助 C 语言程序开发人员在编程时实现易查、易学、易用，并使其成为程序开发必备的工具书。

本书内容

本书按照 C 语言库函数的头文件进行分类，每个头文件为一章内容，分别介绍了对应头文件中常用的 C 语言函数。对每个头文件中的函数按照字母进行排序，其中每个函数都按照功能、语法来介绍，在每个方法后都有相关的示例。

本书特点

☑ 即查、即学、即用

本书将查、学、用结合为一体，读者能够使用本书进行快速查询和快速应用，既是实用的工具书，也是学习的参考书。

☑ 字母排序，快捷搜索

本书在讲解 C 语言函数时按照头文件进行分类，用户在使用时可以根据头文件快速地缩小查找范围，并且每个头文件以字母顺序进行排序，以便读者更快、更准确地查找相关内容。

☑ 内容全面，示例丰富

本书内容涵盖了学习 C 语言应用的各方面技术。为了便于读者快速掌握每个知识点，每个知识点都给出对应的一个或多个示例，并提供了详尽的注释。

☑ 精彩栏目，贴心提醒

本书根据需要在各章使用了“注意”、“说明”和“技巧”等小栏目，让读者可以在学习过程中更轻松地了解相关知识点及概念，并轻松地掌握个别技术的应用技巧。



本书配套资源

本书提供了内容丰富的配套资源，包括源程序、素材，以及模块库、案例库、题库、素材库等多项辅助内容，读者朋友可以通过如下方式获取。

第 1 种方式：

(1) 登录 www.tup.com.cn，在网页右上角的搜索文本框中输入本书书名（注意区分大小写和留出空格），或者输入本书关键字，或者输入本书 ISBN 号（注意去掉 ISBN 号间隔线“-”），单击“搜索”按钮。

(2) 找到本书后单击超链接，在该书的网页下侧单击“网络资源”超链接，即可下载。

第 2 种方式：

访问本书的新浪微博：C 语言图书，找到配套资源的链接地址进行下载。

读者对象

本书适合以下人群阅读。

- | | |
|---|--|
| ☒ 初学编程的自学者 | ☒ 编程爱好者 |
| ☒ 大中专院校的老师和学生 | ☒ 相关培训机构的老师和学员 |
| ☒ 初、中级程序开发人员 | ☒ 程序测试及维护人员 |
| ☒ 参加实习的程序员 | |

读者服务&本书勘误

读者在使用本书过程中遇到的所有问题，均可通过以下方式联系我们。

1. 新浪微博：C 语言图书。

及时发布读者答疑、本书勘误、配套资料更新等内容。

2. 腾讯 QQ：4006751066。

3. 登录网站：www.mingribook.com，在论坛、勘误发布、读者纠错、技术支持、读者之家等栏目中的相关模块中提问、留言或查看。

本书作者

本书由明日科技组织编写，主要编写人员有王雪、赵永发、孙秀梅、曹飞飞、朱晓、陈丹丹、王国辉、张振坤、高春艳、王小科、赵会东、李继业、赛奎春、潘凯华、刘欣、李慧、杨丽、刘龄龄、陈英、肖鑫等。在编写本书的过程中，我们以科学、严谨的态度，力求精益求精，但错误、疏漏之处在所难免，敬请广大读者批评指正。

编 者

目 录

Contents



第 1 章 动态内存管理函数: alloc.h..... 1

- 1.1 brk()——重置数据段存储空间函数..... 2
- 1.2 coreleft()——获取未使用内存大小
函数..... 3
- 1.3 farcalloc()——申请远堆空间函数..... 4
- 1.4 farcoreleft()——获取远堆空闲存储
空间大小函数..... 5
- 1.5 farfree()——释放远堆空间函数..... 6
- 1.6 farmalloc()——申请远堆空间函数..... 6
- 1.7 farrealloc()——远堆存储块空间调整
函数..... 7
- 1.8 sbrk()——增加数据段存储空间函数... 8

第 2 章 DEBUG 相关函数库: assert.h 10

assert()——错误判断函数..... 11

第 3 章 BIOS 相关函数库: bios.h..... 12

- 3.1 bioscom()——串行 I/O 通信函数..... 13
- 3.2 biosdisk()——软硬盘 I/O 函数..... 14
- 3.3 biosequip()——检查设备函数..... 15
- 3.4 bioskey()——键盘接口函数..... 16
- 3.5 biosmemory()——获取存储块大小
函数..... 18
- 3.6 biostime()——设置 bios 时间函数..... 19

第 4 章 控制台输入/输出函数库: conio.h..... 21

- 4.1 cgets()——从控制台输入字符串
函数..... 22
- 4.2 clreol()——清除字符到行末函数..... 22
- 4.3 clrscr()——清除文本模式窗口函数... 23

- 4.4 cprintf()——格式化输出到屏幕
函数..... 24
- 4.5 cputs()——字符串输出到屏幕函数... 25
- 4.6 cscanf()——格式化输入函数..... 26
- 4.7 delline()——删除行函数..... 27
- 4.8 getch()——无回显字符输入函数..... 28
- 4.9 getche()——有回显字符输入函数..... 28
- 4.10 getpass()——以密码形式输入
函数..... 29
- 4.11 gettext()——文本复制函数..... 30
- 4.12 gettextinfo()——获取文本模式的
屏幕信息函数..... 31
- 4.13 gotoxy()——设置光标函数..... 32
- 4.14 highvideo()——高亮显示文本
函数..... 33
- 4.15 insline()——插入空行函数..... 34
- 4.16 kbhit()——检查键盘按键函数..... 35
- 4.17 lowvideo()——低亮显示文本
函数..... 36
- 4.18 movetext()——移动屏幕文本
函数..... 37
- 4.19 normvideo()——正常亮显示文本
函数..... 38
- 4.20 putch()——字符输出函数..... 39
- 4.21 puttext()——文本复制函数..... 40
- 4.22 textattr()——设置文本属性函数..... 41
- 4.23 textbackground()——文本背景色
选择函数..... 42
- 4.24 textcolor()——文本字符颜色选择
函数..... 43



Note

- 4.25 textmode()——文本模式设置函数 43
- 4.26 ungetch()——把字符退回键盘缓冲
区函数 44
- 4.27 wherex()——返回水平光标位置
函数 45
- 4.28 wherey()——返回垂直光标位置
函数 46
- 4.29 window()——定义活动文本窗口
函数 47

第 5 章 字符函数库: ctype.h 49

- 5.1 isascii()——判断字符是否为 ASCII 码
函数 50
- 5.2 isalnum()——判断字符是否为字母或
数字函数 50
- 5.3 isalpha()——判断字符是否为英文
字母函数 51
- 5.4 iscntrl()——判断字符是否为控制
字符函数 52
- 5.5 isdigit()——判断字符是否为十进制
数字函数 53
- 5.6 isgraph()——判断字符是否为除空格
外的可打印字符函数 54
- 5.7 islower()——判断字符是否为小写
英文字母函数 55
- 5.8 isprint()——判断字符是否为可打印
字符(含空格)函数 56
- 5.9 ispunct()——判断字符是否为标点
符号函数 57
- 5.10 isspace()——判断字符是否为空白
字符函数 57
- 5.11 isupper()——判断字符是否为大写
英文字母函数 58
- 5.12 isxdigit()——判断字符是否为十六
进制数字函数 59
- 5.13 toascii()——将字符转换为 ASCII 码
函数 60
- 5.14 tolower()——将字符转换为小写
英文字母函数 61

- 5.15 toupper()——将字符转换为大写
英文字母函数 62

第 6 章 目录操作函数库: dir.h 63

- 6.1 chdir()——更改当前工作目录函数 64
- 6.2 findfirst()——搜索当前磁盘目录
函数 65
- 6.3 findnext()——搜索磁盘下一个匹配的
文件函数 66
- 6.4 fnmerge()——构造完整路径函数 66
- 6.5 fnsplit()——拆分路径函数 67
- 6.6 getcurdir()——获取指定驱动器上的
工件目录函数 69
- 6.7 getcwd()——获取当前工作目录
函数 69
- 6.8 getdisk()——获取当前磁盘驱动器号
函数 70
- 6.9 mkdir()——建立目录函数 71
- 6.10 mktemp()——生成不重复文件名
函数 72
- 6.11 rmdir()——删除文件目录函数 73
- 6.12 setdisk()——设置当前工作驱动器
函数 73
- 6.13 searchpath()——查找文件函数 74

第 7 章 系统接口函数库: dos.h 76

- 7.1 allocmem()——DOS 分配存储段
函数 77
- 7.2 bdos()——DOS 系统调用函数 78
- 7.3 country()——返回国家相关信息
函数 79
- 7.4 ctrlbrk()——设置 Ctrl-Break 处理
程序函数 80
- 7.5 delay()——以毫秒为单位的延时
函数 81
- 7.6 doserror()——获取扩展 DOS 错误
信息函数 82
- 7.7 dostounix()——把 DOS 时间转换为
UNIX 时间 83



7.8	FP_OFF()——获取远指针的偏 移量	84
7.9	FP_SEG()——获取远指针的段 地址	85
7.10	freemem()——释放已分配 DOS 内 存块函数	86
7.11	getcbrk()——获取 Ctrl-Break 设置 函数	87
7.12	getdate()——获取 DOS 日期函数	88
7.13	getdta()——获取磁盘传输地址 函数	89
7.14	gettime()——获取系统时间函数	90
7.15	inport()——从硬件端口中输入一个 字函数	91
7.16	inportb()——从硬件端口中输入 一个字节函数	92
7.17	int86()——通用 8086 软中断接口 函数	93
7.18	int86x()——增强型通用 8086 软 中断接口函数	94
7.19	intdos()——通用 DOS 中断接口 函数	95
7.20	intdosx()——增强型通用 DOS 中断 接口函数	97
7.21	intr()——改变软中断接口函数	98
7.22	keep()——退出并继续驻留函数	99
7.23	MK_FP()——远指针合成	100
7.24	nosound()——关闭 PC 扬声器 函数	101
7.25	outport()——从硬件端口中输出 一个字函数	102
7.26	outportb()——从硬件端口中输出 一个字节函数	103
7.27	parsfnm()——分析文件名函数	104
7.28	peek()——从存储单元读一个字 函数	105
7.29	peekb()——从存储单元读一个字 节的函数	106

7.30	poke()——存一个字到给定存储 单元函数	108
7.31	pokeb()——存一个字节到给定 存储单元函数	109
7.32	randbrd()——随机块读函数	110
7.33	randbwr()——随机块写函数	111
7.34	segread()——读段寄存器值函数	112
7.35	setblock()——修改 DOS 分配地址 函数	113
7.36	setcbrk()——设置 Ctrl-Break 函数	114
7.37	setdate()——设置 DOS 日期函数	115
7.38	settime()——设置 DOS 时间函数	116
7.39	setdta()——设置磁盘传输地址 函数	117
7.40	sleep()——以秒为单位的延时 函数	118
7.41	sound()——打开 PC 扬声器函数	119
7.42	unixtodos()——把 UNIX 时间转换 为 DOS 时间函数	120

第 8 章 浮点数据处理函数库:

float.h	122
8.1 _clear87()——清除状态字函数	123
8.2 _control87()——设置控制字函数	124
8.3 _status87()——获取浮点处理器状态 值函数	125

第 9 章 图形处理函数库:

graphics.h	126
9.1 arc()——画弧线函数	127
9.2 bar()——绘制二维条形图函数	128
9.3 bar3d()——绘制三维条形图函数	129
9.4 circle()——画圆函数	130
9.5 cleardevice()——清除图形屏幕 函数	131
9.6 clearviewport()——清除图形视区 函数	132
9.7 closegraph()——关闭图形系统	



Note



- 函数 133
- 9.8 detectgraph()——通过检测硬件确定图形驱动程序和模式函数 134
- 9.9 drawpoly()——绘制多边形函数 134
- 9.10 ellipse()——绘制椭圆函数 135
- 9.11 fillellipse()——绘制并填充椭圆函数 137
- 9.12 fillpoly()——绘制多边形并填充函数 138
- 9.13 floodfill()——填充有界区域函数 139
- 9.14 getarccoords()——取得最后一次调用画弧线坐标函数 140
- 9.15 getspectratio()——获取当前图形模式的纵横比函数 141
- 9.16 getbkcolor()——获取当前背景颜色函数 142
- 9.17 getcolor()——获取当前画线颜色函数 143
- 9.18 getdefaultpalette()——返回默认调色板函数 144
- 9.19 getdrivername()——获取图形驱动程序名函数 145
- 9.20 getfillpattern()——获取用户定义的填充模式函数 146
- 9.21 getfillsettings()——获取填充模式和填充颜色函数 147
- 9.22 getgraphmode()——获取当前图形模式函数 148
- 9.23 getimage()——将指定区域的位图调入内存函数 149
- 9.24 getlinesettings()——获取当前线型、模式和宽度函数 150
- 9.25 getmaxcolor()——获取最大颜色函数 151
- 9.26 getmaxmode()——获取当前驱动程序的最大图形模式 152
- 9.27 getmaxx()——返回屏幕的最大x 坐标函数 153
- 9.28 getmaxy()——返回屏幕的最大y 坐标函数 154
- 9.29 getmodename()——获取指定图形模式的名称 155
- 9.30 getmoderange()——获取给定图形驱动程序的模式范围函数 156
- 9.31 getpixel()——获取指定像素的颜色函数 157
- 9.32 getpalette()——获取调色板函数 158
- 9.33 getpalettesize()——获取调色板的颜色数函数 159
- 9.34 gettextsettings()——获取当前图形文本字体的信息函数 160
- 9.35 getviewsettings()——获取当前视口的信息函数 161
- 9.36 getx()——获取当前图形位置的x 坐标函数 163
- 9.37 gety()——获取当前图形位置的y 坐标函数 164
- 9.38 graphdefaults()——将所有图形设置复位为它们的默认值函数 164
- 9.39 grapherrormsg()——返回一个错误信息串的指针函数 165
- 9.40 _graphfreemem()——用户可修改的图形存储区释放函数 166
- 9.41 _graphgetmem()——用户可修改的图形存储区分配函数 167
- 9.42 graphresult()——返回最后一次不成功的图形操作的错误代码函数 168
- 9.43 imagesize()——返回保存位图像所需的字节数函数 169
- 9.44 initgraph()——初始化图形系统函数 170
- 9.45 line()——在指定两点间画直线函数 172
- 9.46 linerel()——从当前位置点到相对位移点绘制直线函数 173



9.47	lineto()——在从当前位置点到指定点间画直线函数	174
9.48	moverel()——将当前点移动到一相对距离函数	175
9.49	moveto()——将当前点移动到(x,y)函数	176
9.50	outtext()——在视区显示字符串函数	176
9.51	outtextxy()——在指定位置显示字符串函数	177
9.52	pieslice()——绘制扇形并填充函数	178
9.53	putimage()——在屏幕上输出位图函数	179
9.54	putpixel()——在指定位置绘制像素函数	180
9.55	rectangle()——绘制矩形函数	181
9.56	restorecrtmode()——将屏幕模式恢复为先前的 initgraph 设置函数	182
9.57	sector()——绘制并填充椭圆扇区函数	183
9.58	setactivepage()——设置图形输出活动页函数	184
9.59	setallpalette()——按指定方式改变所有的调色板颜色函数	185
9.60	setaspectratio()——设置图形纵横比函数	187
9.61	setbkcolor()——设置当前背景颜色函数	187
9.62	setcolor()——设置当前画线颜色函数	188
9.63	setfillpattern()——选择用户定义的填充模式函数	189
9.64	setfillstyle()——设置填充模式和颜色函数	191
9.65	setgraphmode()——设置当前图形模式函数	192
9.66	setlinestyle()——设置当前画线宽度	

	和类型函数	193
9.67	setpalette()——设置有关当前调色板的信息函数	193
9.68	settextjustify()——设置文本的对齐方式函数	194
9.69	settextstyle()——设置当前的文本属性函数	195
9.70	setusercharsize()——为矢量字体改变字符宽度和高度函数	196
9.71	setviewport()——为图形输出设置当前视口函数	197
9.72	setvisualpage()——设置可见图形页号函数	198
9.73	setwritemode()——设置输出模式	198
9.74	textheight()——获取以像素为单位的字符串高度函数	200
9.75	textwidth()——获取以像素为单位的字符串宽度函数	201

第 10 章 输入/输出函数库: io.h..... 203

10.1	access()——文件访问权限设置函数	204
10.2	chmod()——改变文件访问方式函数	205
10.3	close()——关闭文件函数	206
10.4	creat()——创建文件函数	206
10.5	dup()——复制文件句柄函数	208
10.6	eof()——文件检测结束函数	209
10.7	fdopen()——连接文件句柄函数	210
10.8	filelength()——文件字节数统计函数	211
10.9	getftime——获取文件时间函数	211
10.10	isatty()——设备类型检查函数	212
10.11	lock()——文件共享锁设置函数	213
10.12	lseek()——移动文件指针函数	214
10.13	open()——打开文件函数	216
10.14	read()——读文件函数	217





- 10.15 setmode()——设置文件打开方式
函数 218
- 10.16 tell()——获取文件指针位置
函数 219
- 10.17 unlock()——解除共享锁函数 219
- 10.18 write()——写文件函数 220

第 11 章 数学函数库: math.h 221

- 11.1 abs()——整数绝对值函数 222
- 11.2 acos()——反余弦值函数 223
- 11.3 asin()——反正弦值函数 224
- 11.4 atan()——反正切值函数 225
- 11.5 atan2()——X/Y 反正切函数 226
- 11.6 ceil()——不小于某数的最小整数
函数 226
- 11.7 cos()——余弦值函数 227
- 11.8 cosh()——双曲余弦值函数 228
- 11.9 exp()——e 的次幂函数 229
- 11.10 fabs()——浮点数绝对值函数 230
- 11.11 floor()——不大于某数的最大整数
函数 231
- 11.12 fmod()——余数函数 232
- 11.13 frexp()——计算浮点数尾数和指数
函数 233
- 11.14 hypot()——计算直角三角形斜边
长度函数 234
- 11.15 labs()——长整型绝对值函数 235
- 11.16 ldexp()——计算幂函数 235
- 11.17 log()——自然对数函数 236
- 11.18 log10()——对数函数 237
- 11.19 modf()——取浮点数小数部分
函数 238
- 11.20 pow()——计算 x 的 y 次幂函数 ... 239
- 11.21 pow10()——计算 10 的 x 次幂
函数 240
- 11.22 sin()——正弦值函数 241
- 11.23 sinh()——双曲正弦值函数 242
- 11.24 sqrt()——平方根函数 243
- 11.25 tan()——正切值函数 244

- 11.26 tanh()——双曲正切值函数 245

第 12 章 内存相关函数库: mem.h 246

- 12.1 memcpy()——字节复制函数 247
- 12.2 memchr()——字符搜索函数 248
- 12.3 memcmp()——串比较函数 249
- 12.4 memcopy()——字节复制函数 250
- 12.5 memicmp()——串比较函数 251
- 12.6 memmove()——字节移动函数 252
- 12.7 memset()——串设置函数 253

第 13 章 进程管理函数库:

process.h 254

- 13.1 spawnl()——创建并运行子程序
函数 255
- 13.2 spawnle()——创建并运行子程序
函数 256

第 14 章 函数跳转函数库:

setjmp.h 258

- 14.1 longjmp()——非局部转移执行
函数 259
- 14.2 setjmp()——非局部转移函数 260

第 15 章 标准库函数库: stdarg.h 262

- 15.1 va_arg()——调用可变参数列表
函数 263
- 15.2 va_end()——停止使用可变参数
函数 264
- 15.3 va_start()——开始使用可变参数
函数 265
- 15.4 vfprintf()——格式化输出流函数 ... 266
- 15.5 vfscanf()——执行流中格式化输入
函数 267
- 15.6 vprintf()——格式化输出函数 269
- 15.7 vsprintf()——格式化输出串函数 ... 270
- 15.8 vscanf()——执行格式化输入
函数 271
- 15.9 vsscanf()——执行流中格式化输入
函数 272



第 16 章 输入/输出库: `stdio.h` 274

- 16.1 `clearerr()`——复位错误标志函数 275
- 16.2 `fclose()`——关闭流函数 276
- 16.3 `feof()`——检测流上的文件结束符
函数 276
- 16.4 `ferror()`——检测流上的错误
函数 277
- 16.5 `fflush()`——清除流函数 278
- 16.6 `fgetc()`——从流中读取字符函数 279
- 16.7 `fgetchar()`——从流中读取字符
函数 280
- 16.8 `fgetpos()`——取得当前文件句柄
函数 281
- 16.9 `fgets()`——从流中读取字符函数 282
- 16.10 `fopen()`——打开流函数 283
- 16.11 `fprintf()`——将格式化内容输出
到流函数 284
- 16.12 `fputc()`——输出字符到标准
输出流 (`stdout`) 函数 285
- 16.13 `fputs()`——送字符串到流中
函数 286
- 16.14 `fread()`——从流中读数据函数 287
- 16.15 `freopen()`——替换流函数 288
- 16.16 `fscanf()`——从流中执行格式化
输入内容函数 289
- 16.17 `fseek()`——重定位流上的文件
指针函数 290
- 16.18 `fsetpos()`——定位流上的文件
指针函数 291
- 16.19 `ftell()`——返回当前文件指针
函数 292
- 16.20 `fwrite()`——写内容到流函数 293
- 16.21 `getc()`——从流中取字符函数 294
- 16.22 `getchar()`——从 `stdin` 流中读字符
函数 295
- 16.23 `getche()`——从控制台取字符
(带回显) 函数 296
- 16.24 `gets()`——从流中取字符函数 297

- 16.25 `getw()`——从流中取整数函数 298
- 16.26 `perror()`——系统错误信息 299
- 16.27 `printf()`——格式化输出函数 300
- 16.28 `putc()`——输出一个字符到流中
函数 301
- 16.29 `putchar()`——在 `stdout` 上输出
字符函数 301
- 16.30 `puts()`——将字符串送到流中
函数 302
- 16.31 `putw()`——将字符或字送到流中
函数 303
- 16.32 `rename()`——重命名文件函数 305
- 16.33 `remove()`——删除文件函数 305
- 16.34 `rewind()`——将文件指针重新指向
流的开头函数 306
- 16.35 `scanf()`——格式化输入函数 307
- 16.36 `setbuf()`——把缓冲区与流相联
函数 308
- 16.37 `setvbuf()`——把缓冲区与流相关
函数 309
- 16.38 `sprintf()`——将格式化数据输出到
字符串中函数 310
- 16.39 `sscanf()`——执行字符串中的格式化
输入函数 311
- 16.40 `tmpfile()`——以二进制方式打开
暂存文件函数 312
- 16.41 `tmpnam()`——创建一个唯一的
文件名函数 313
- 16.42 `ungetc()`——把字符退回到输入
流函数 314
- 16.43 `ungetch()`——把字符退回到键盘
缓冲区函数 315

第 17 章 标准工具库函数库:

`stdlib.h` 317

- 17.1 `abort()`——异常终止进程函数 318
- 17.2 `atexit()`——注册终止函数 318
- 17.3 `atof()`——转换字符串函数 319
- 17.4 `atoi()`——转换字符串函数 320



17.5	atol()——转换字符串函数	321
17.6	bsearch()——二分法搜索函数	322
17.7	calloc()——分配主存储器函数	323
17.8	div()——整数相除函数	324
17.9	ecvt()——转换浮点数函数	325
17.10	exit()——终止程序函数	326
17.11	fcvt()——转换浮点数函数	327
17.12	free()——释放已分配块函数	328
17.13	gcvt()——转换浮点数函数	329
17.14	getenv()——获取环境中字符串 函数	330
17.15	itoa()——转换整数函数	331
17.16	ldiv()——长整型数相除函数	332
17.17	lfind()——线性搜索函数	333
17.18	lsearch()——线性搜索函数	334
17.19	malloc()——内存分配函数	335
17.20	putenv()——加载环境中字符串 函数	336
17.21	qsort()——快速排序函数	338
17.22	rand()——随机数发生器函数	339
17.23	random()——随机数发生器 函数	340
17.24	randomize()——随机数发生器 函数	340
17.25	realloc()——重新分配主存函数 ...	341
17.26	srand()——初始化随机数发生器 函数	342
17.27	strtod()——转换字符串函数	343
17.28	strtol()——转换字符串函数	344
17.29	swab()——交换字节函数	345
17.30	system()——发出 DOS 命令 函数	346

第 18 章 信号定义函数库:

signal.h		347
18.1	raise()——发送信号函数	348
18.2	signal()——设置信号对应动作 函数	348

第 19 章 字符串函数库: string.h 350

19.1	movmem()——从源字符串中移动字节 到目标字符串函数	351
19.2	setmem()——把内存区域的字节 设置成字符函数	352
19.3	strcpy()——把字符串复制到数组 函数	353
19.4	strcat()——字符串追加函数	354
19.5	strchr()——查找字符串首次出现 位置函数	355
19.6	strcmp()——字符串比较函数	356
19.7	strcmapi()——字符串比较（不区分 大小写）函数	357
19.8	strncpy()——复制字符串到数组 函数	357
19.9	strcspn()——字符串查找函数	358
19.10	strdup()——字符串复制函数	359
19.11	stricmp()——字符串比较（不区分 大小写）函数	360
19.12	strlen()——字符串长度函数	361
19.13	strlwr()——将字符串转换为小写 形式函数	362
19.14	strncat()——字符串尾部追加 函数	362
19.15	strncmp()——字符串比较函数	363
19.16	strncmapi()——字符串比较（不区 分大小写）函数	364
19.17	strncpy()——将字符串复制到数组 函数	365
19.18	strnicmp()——字符串比较（不区分 大小写）函数	366
19.19	strpbrk()——字符串查找函数	367
19.20	strrev()——字符串倒序函数	368
19.21	strset()——将字符串设置成指定 字符函数	369
19.22	strstr()——在字符串中查找指定 字符首次出现位置函数	370
19.23	strtok()——用分隔符分解字符串	





函数	371	串函数	376
19.24strupr()——将字符串转换为大写		20.4difftime()——计算时间差函数	376
形式函数	372	20.5gmtime()——获取格林尼治结构的	
第 20 章 时间函数库: time.h	373	时间函数	377
20.1asctime()——日期和时间转换		20.6localtime()——获取 tm 结构的系统	
函数	374	时间函数	378
20.2clock()——测定运行时间函数	375	20.7time()——获取或设置时间函数	379
20.3ctime()——转换日期和时间为字符		20.8tzset()——UNIX 时间兼容函数	380



Note

第1章

动态内存管理函数：alloc.h

动态内存管理函数在程序执行过程中为程序中的数据分配内存，这些函数都通过指针来指明内存地址。

最常用的动态内存管理函数是 `stdlib` 中的 `malloc()`、`calloc()` 和 `free()`，这组函数在同一数据段中分配内存，当程序中动态分配的内存很多，以至于跨越数据段分配时，用到本章中的函数，其实用性稍差。

本程序不能用 VC 编译，所有程序均在 TC 2.0 中编译运行。



1.1 brk()——重置数据段存储空间函数



Note

brk() 函数

实用程度：★★☆☆☆

实例位置：配套资源\SL\01\01

brk()函数用于通过重新设置数据段结束位置，改变数据段存储空间大小。

语法

```
int brk(void *endds);
```

brk()函数的语法参数说明如下：

参数 endds 为重新设置的数据段结束地址。

brk()函数的返回值：函数成功执行时返回值为 0，不成功则为-1。

示例

本示例演示读取文本文件中的所有整数。使用 malloc()分配 MAX 个整数空间，用于保存数据，执行中如果 MAX 个空间用完而文件中还有未读完的数据，则用 brk()函数重新设置数据段存储空间。

程序执行前先在 D:\下用记事本建立文件 aa.txt，输入 1 2 3 4 5 6 7 8（每个数之间用一个空格分开）。其具体代码如下：

```
#include "stdio.h"
#include "alloc.h"
#define MAX 500                                /*预定义空间，也是每次增加的空间容量*/
main()
{
    int *p,n=0,max=MAX;                        /*max 实际已分配的空间*/
    FILE *fp;
    clrscr();                                  /*清屏*/
    fp=fopen("d:\\aa.txt","r");                /*打开文件*/
    p=(int*)malloc(MAX*sizeof(int));           /*预先分配 MAX 个整数空间*/
    while(!feof(fp))                           /*只要没到文件尾就从文件输入*/
    {
        if(n==max)                             /*空间用完时，增加数据段大小*/
        {
            max=max+MAX;
            brk(p+max);                          /*将数据段结束位置定在 max+MAX，即增加 MAX*/
        }
        fscanf(fp,"%d",&p[n]);                 /*从文件读一个整数*/
        printf("%d ",p[n]);                     /*显示这个数*/
        n++;                                     /*实际读出数据个数*/
    }
}
```



```

    }
    fclose(fp);           /*关闭文件*/
    free(p);              /*释放内存*/
}

```

运行结果如图 1.1 所示。

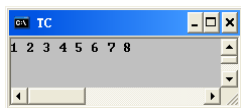


图 1.1 brk()函数的执行结果（文件中只有 8 个数）



Note

1.2 coreleft()——获取未使用内存大小函数

coreleft()函数

实用程度：★★☆☆☆

实例位置：配套资源\SL\01\02

coreleft()函数用于获取未使用内存空间大小。

语法

unsigned coreleft(void):

coreleft()函数没有参数。

coreleft()函数的返回值：未使用内存的大小。

示例

本示例演示通过 brk()函数改变动态分配空间大小，在改变前后用 coreleft()函数输出剩余内存空间。其具体代码如下：

```

#include "stdio.h"
#include "alloc.h"
main()
{
    char *ptr;
    clrscr();           /*清屏*/
    printf("Chang data segment space allocation with brk()\n");
    ptr=malloc(4);      /*分配 4 个字节， ptr 为内存地址数据段起始地址*/
    printf("Before brk() call:%lu bytes free\n",coreleft());    /*coreleft()为未使用内存大小*/
    brk(ptr+1000);      /*数据段结束位置设置为 prt+1000*/
    printf("After brk() call:%lu bytes free\n",coreleft());
    free(ptr);          /*释放内存*/
}

```

运行结果如图 1.2 所示。